UCRL- 86804
PREPRINT

# CIRCUIT ANALYSIS AND SIMULATION PROGRAMS
# -AN OVERVIEW-

W. G. Magnuson, Jr.

IEEE 1981
Nuclear Science Symposium
San Francisco, CA
October 21-23, 1981

October 1981

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

Mexico issued a contract with IBM in the summer of 1962 to develop a general-purpose circuit analysis program. This program, called PREDICT, was to have features in it for the determination of circuit responses in radiation environments. PREDICT was released in the summer of 1964 [5].

TAP also initiated the development of the ECAP program at IBM. ECAP became available in the fall of 1964 and became a standard in industry and universities for comparing circuit analysis programs [6]. Because Branin had worked at the Los Alamos Scientific Laboratory in the mid-1950's, a program development started there which produced the NET-1 program in 1964 [7]. During the development of NET-1 a summer visit by R. Dickhaut of Boeing Aircraft to Los Alamos initiated the start of the CIRCUS program [8].

These programs (PREDICT, NET-1, and CIRCUS) all were aimed at analyzing electronic circuits subjected to a radiation environment. In the mid-1960's several other programs also appeared on the scene (CIRCAL, CORNAP, CALAHAN, LISA) and represented an era of different analysis methods and different numerical techniques being used. Nodal, state variable, topological, and hybrid network formulation techniques were employed.

In the late 1960's and early 1970's several improvements in numerical and matrix methods were incorporated into analysis programs. Implicit algorithms for solving networks characterized by stiff state equations appeared [9]. These algorithms essentially solved one of the problems which had plagued early analysis programs-networks with a wide spread in circuit time constants. Sparse matrix methods appeared and were employed to deal with larger networks and to speed solutions [10].

These improvements led to the second generation of programs such as ECAP-II, SCEPTRE, SCEPTRE-II, ASTAP, NET-2, CIRCUS-2, SPICE, and others [11, 12, 13, 14, 15, 16, 17]. Most of the severe technical limitations of the earlier programs have been overcome. Problems remain in modeling and in size as integrated circuits grow in device count.

## The Anatomy of an Analysis Program

Chua and Lin, in their book, Computer-Aided Analysis of Electronic Circuits [18], have a good description of what makes up a general-purpose analysis program. Basically there are five stages:

1. Input stage - which reads the input, checks for syntax, and stores topology and element information.

2. Device model replacement stage - where either built-in models are used or an external model library is read and stored into an array.

3. Equation formulation stage - where the program formulates the equations for the network response. There are several methods used and several types of analyses which can be performed.

4. The solution stage - where the equations of stage 3 are solved.

5. The output stage - where output becomes available to the user.

Many human factor features are desirable for the input of circuit and element information to programs. Because the input generally comes from schematics and is done by humans (but not always) we would like a field-free format which can be easily learned and is understandable. Almost all programs have such an input language. Because frequently we are dealing with nonlinear elements, we would like a way of describing equations, tables, and functional relationships. Some of the programs have these capabilities. Likewise we wish error detection and reporting to inform us of problem areas or mistakes in the input. Some of the analysis programs give help along these lines.

At the modeling stage, desirable features include built-in models (BJT, FET, MOSFET, etc.), multilevel nesting (networks containing subnetworks), and capability for adding external models (for new or unique devices). Some programs have extensive modeling capability.

The three major analysis modes are: dc analysis, ac analysis, and transient analysis. A dc analysis is the steady-state solution of the circuit. The dc solution generally becomes the starting point for ac and transient analyses. An ac analysis is a small-signal steady-state solution with all nonlinearities linearized about the circuit's operating point. The input for an ac analysis is generally a swept frequency to give a frequency spectrum of the response. A transient analysis computes the time response to inputs. The inputs can be steps, pulses, sinusoidal, or other complex waveforms.

Besides the three above major solutions other types of analyses can be performed. For example some of the major analysis programs can perform a sensitivity analysis, a worst case analysis, or a Monte Carlo analysis. A sensitivity analysis computes a change in an output variable with respect to changes in other network parameters. A worst-case analysis computes the bounds in output response when elements take on their extreme limits in values. A Monte Carlo analysis computes responses for a series of randomly chosen values for elements with tolerance limits. The Monte Carlo output is usually a histogram.

Other types of analyses performed by some programs are: noise analysis, temperature analysis, distortion analysis, and transfer-characteristic analysis.

The network equations which are solved by a circuit analysis program are formed from the input element-interconnection (network topology). A number of formulations have been used by different analysis programs. Some of the more common techniques and representative programs are shown in Figure 3.

Tableau - ASTAP, NET-2, CIRCUS-2
State Variable - SCEPTRE-2, BELAC, CORNAP, PREDICT, CIRCUS, NET-1
Hybrid - ECAP-II
Nodal - SPICE, CIRC, SYSCAP, AEDCAP, LISA, TRAC, ECAP
Topological - SNAP, NASAP, CALAHAN

Figure 3
Circuit Analysis Formulation Methods and Examples

The tableau formulation method represents Kirchhoff's voltage and current laws and component functional relationships in sparse matrix form. This sparse matrix or tableau is then operated directly upon in the solution process, usually by a form of Gaussian elimination. The method is usually an improvement in both reducing the number of calculations (speed) and

# CIRCUIT ANALYSIS AND SIMULATION PROGRAMS
# -AN OVERVIEW-

W. G. Magnuson, Jr.
Lawrence Livermore National Laboratory
P.O. Box 5504, L-156
Livermore, California 94550

## Summary

The paper is intended to be a tutorial-like description of the development of electrical circuit analysis programs used for analyzing digital and analog circuits. The anatomy of circuit simulator programs, analysis formulation techniques, and solution procedures will be addressed. A few of the features, capabilities, and limitations of several of the more widely used current programs will also be covered.

## The Need for Circuit Analysis

As engineering systems have become larger, increasingly complex, and more costly, the designer has turned to relying on more sophisticated design techniques. These techniques have, to an increasing amount, centered on the use of the digital computer as the central design and analysis tool. What is design? Design can be thought of as iterative analysis until certain specifications are met. The design process may be characterized by a simple block diagram (Figure 1).

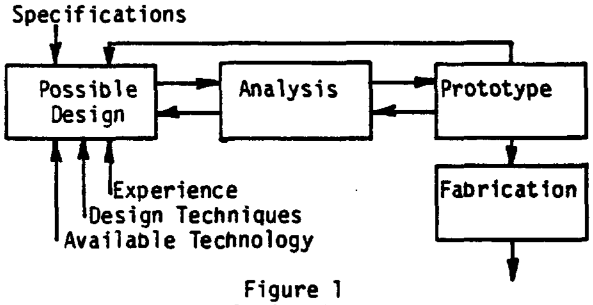


Figure 1
The Design Process

From available technology, design techniques, and experience, the design engineer formulates a possible design to meet the specifications. This design is analyzed to see if the proposed design is reasonable. Feedback occurs at the analysis and prototype (and possibly even includes the specifications) until the final design is achieved.

Of course the above figure represents a greatly simplified picture of some of the interactions. What is intended to be shown is the fundamental role analysis plays in the development of a system. There are many levels of abstraction during the design process and analysis could be considered at each or all levels. For example we could consider system level simulation, register level simulation, gate level simulation, circuit level simulation, or device level simulation if we were considering logic system design. In this paper the focus is on computer programs for circuit analysis.

It is interesting to note the growth of electronics at the circuit level in terms of what has been delivered in a single package (Figure 2). The growth depicted in Figure 2 has had a direct influence on the development of circuit analysis programs.

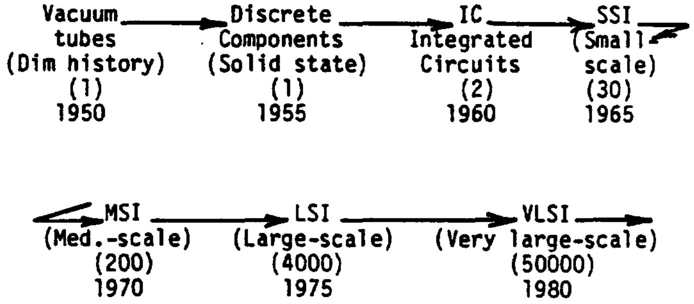

Vacuum tubes (Dim history) (1) 1950
Discrete Components (Solid state) (1) 1955
IC Integrated Circuits (2) 1960
SSI (Small scale) (30) 1965
MSI (Med.-scale) (200) 1970
LSI (Large-scale) (4000) 1975
VLSI (Very large-scale) (50000) 1980

Figure 2
Growth of Solid-state Circuits

In analog circuit design in the 1960's, the utilization of circuit analysis programs as a design aid did not meet with wide success. Analysis programs did find success in design reviews, reliability analyses, and radiation hardening studies. When the transition from MSI to LSI occurred in the early 1970's analog circuit analysis became a vital part of design primarily because of the cost and time to design the LSI diffusion masks.

Part of the early failure of analysis programs to help the designer stemmed from an oversell on the part of the program implementors ("CAD will end your breadboarding"). Failure also occurred from numerical problems in the programs themselves which severely restricted their usefulness.

## The Development of Circuit Analysis Programs

Prior to 1960 computer programs to analyze electrical circuits were either very specific programs to solve for a particular topology or a few more general filter design and analysis programs. The first type was essentially solving response equations for a specific circuit topology. The foundations for analyzing a more general network were laid much earlier by Gabriel Kron in his 1939 book, Tensor Analysis of Networks [1]. Kron's book presented Kirchhoff's laws and network topology as matrices which was an almost ideal representation for digital computer implementation. Branin in the late 1950's and early 1960's employed Kron's methods to develop the program TAP (Transistor Analysis Program) which was one of the first generation programs (1962) [2]. Branin was primarily the one responsible for implementing the matrix algebraic topological techniques of Kron [3]. While Branin (at IBM) was developing TAP, Ashcraft and Hochwalt at Autonetics were developing SPARC and SCAN which resulted in the general program TRAC in 1963 [4].

I think because of the development of TAP at IBM, the Air Force Weapons Laboratory at Albuquerque, New

*Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract number W-7405-ENG-48.

the computational errors (accuracy) over full matrix methods.

The state variable method is based on reducing the network describing equations into a set of first-order differential equations. The capacitor voltages and inductor currents are usually used as dependent variables or state variables. The algebraic equations associated with the network are eliminated by substitution.

The hybrid formulation method is similar to the state variable method with the exception that the algebraic equations are not eliminated.

The nodal method simply uses Kirchhoff's current law to form a set of node equations. Inductor currents and capacitor voltages are solved by difference equations and again a form of Gaussian elimination is used in the solution process as well as sparse matrix techniques for storage.

The topological method using signal flow graph or tree-enumeration methods was used in a few early programs. These approaches have problems of being able only to analyze relatively small networks due to combinational problems. Thus this method has not been used in more modern programs.

The numerical solution process is closely tied to the network formulation type. For linear ac analyses generally a form of complex Gaussian elimination is used. For linear time-domain analysis either numerical integration or, less frequently, the evaluation of the matrix exponential is used. Both Newton-Raphson and piecewise-linear methods are used for evaluating nonlinear dc analyses. For nonlinear transient analyses both explicit and implicit numerical integration methods are in use.

Finally the output stage gives the user the answers from the analysis. Printed tabular output, plots, listings, etc., are all used and depend on the particular analysis performed. An example later will show typical output from an analysis.

## Features and Capabilities of a Few Programs

This section will compare some of the features of a few of the circuit analysis programs in use at Lawrence Livermore National Laboratory. The main programs being used are the SCEPTRE, NET-2, SPICE-2, and EMTP programs. All of these programs are in the public domain and while they run on CDC 7600 and Cray-1 computers at LLNL, versions of the programs exist for other computer systems.

Figure 4 summarizes some of the characteristics of these programs.

| | SCEPTRE | NET-2 | SPICE2 | EMTP | Extended SCEPTRE |
|---|---|---|---|---|---|
| Formulation | State space | Tableau | Nodal | Nodal | State space |
| Transient Integration | Gear's | Implicit trapezoidal | Implicit fixed step trapezoidal | Implicit fixed step trapezoidal | Gear's |
| ac Small Signal | No | Yes | Yes | No | Yes |
| User-defined Models | Yes | No | No | No | Yes |
| Model Nesting | Yes | Yes | Yes | No | Yes |
| Parameter Variation | Yes (limited) | Yes | Temperature | No | Yes |
| Network Optimization | No | Yes | Sensitivity | No | Sensitivity |
| Monte Carlo Analysis | No | Yes | No | No | Yes |

Figure 4
Summary of Program Features

The distinction between SCEPTRE and Extended SCEPTRE lies in the historical development of the SCEPTRE program. In the early 1970's a contract was let for the enhancement of SCEPTRE in which additional analysis features were added to the program. The resulting version called Extended SCEPTRE (and sometimes SCEPTRE-II) became operational on the IBM 360 computers. Extended SCEPTRE was added to and appeared as SUPER*SCEPTRE which is currently available commercially. Extended SCEPTRE is being converted to the Cray-1 computer and should become available early in 1982.

The EMTP (ElectroMagnetic Transients Program) is included in the list because of its use at LLNL [19]. EMTP was developed within the electrical power industry for analyzing distribution networks and, as such, has several useful modeling features (multiphase Pi-equivalent, circuit breakers, and lightning arrestors to name a few). EMTP has proven useful in study high power distribution networks.

The power of SPICE-2 rests in its sophisticated modeling of semiconductor models. The program, developed at the University of California, does a very good job in meeting student needs for low-cost solutions and ease of input. Some generality has been sacrificed to meet these needs however. The program lacks provisions for nonlinear element input and, because of fixed integration step size is used, solution errors can develop. The program is widely used in both universities and industry and is frequently used for analyzing integrated circuit designs of appreciable complexity.

To illustrate the flavor of the input and output a very simple example is shown in Figures 5 and 6. Figure 5(a) shows a circuit schematic for a low pass active filter using an operational amplifier. The SPICE-2 and NET-2 input are shown in Figure 5(b). Figure 6(a) and (b) shows the output for this analysis and is typical of most circuit analysis programs. Only a portion of the output is included.

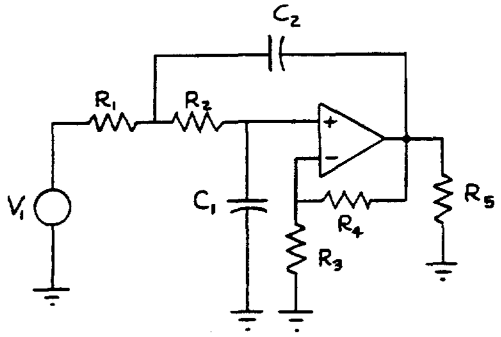
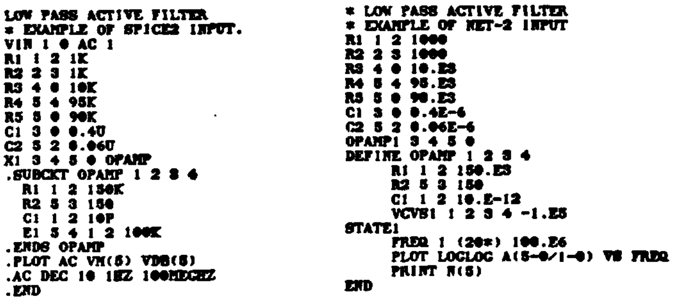


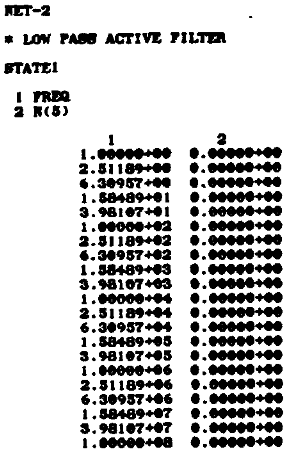
Figure 5(a)
Low Pass Active Filter



```
LOW PASS ACTIVE FILTER
* EXAMPLE OF SPICE2 INPUT.
VIN 1 0 AC 1
R1 1 2 1K
R2 2 3 1K
R3 4 0 10K
R4 5 4 95K
R5 5 0 90K
C1 3 0 0.4U
C2 5 2 0.06U
X1 3 4 5 0 OPAMP
.SUBCKT OPAMP 1 2 3 4
  R1 1 2 150K
  R2 5 3 150
  C1 1 2 10P
  E1 5 4 1 2 100K
.ENDS OPAMP
.PLOT AC VM(5) VDB(5)
.AC DEC 10 1HZ 100MEGHZ
.END
```

```
* LOW PASS ACTIVE FILTER
* EXAMPLE OF NET-2 INPUT
R1 1 2 1000
R2 2 3 1000
R3 4 0 10.E3
R4 5 4 95.E3
R5 5 0 90.E3
C1 3 0 0.4E-6
C2 5 2 0.06E-6
OPAMP1 3 4 5 0
DEFINE OPAMP 1 2 3 4
     R1 1 2 150.E3
     R2 5 3 150
     C1 1 2 10.E-12
     VCVS1 1 2 3 4 -1.E5
STATE1
     FREQ 1 (20*) 100.E6
     PLOT LOGLOG A(5-0/1-0) VS FREQ
     PRINT N(5)
END
```

Figure 5(b)
SPICE-2 and NET-2 Input



```
NET-2

* LOW PASS ACTIVE FILTER

STATE1

 1 FREQ
 2 N(5)

             1             2
       1.00000+00   0.00000+00
       2.51189+00   0.00000+00
       6.30957+00   0.00000+00
       1.58489+01   0.00000+00
       3.98107+01   0.00000+00
       1.00000+02   0.00000+00
       2.51189+02   0.00000+00
       6.30957+02   0.00000+00
       1.58489+03   0.00000+00
       3.98107+03   0.00000+00
       1.00000+04   0.00000+00
       2.51189+04   0.00000+00
       6.30957+04   0.00000+00
       1.58489+05   0.00000+00
       3.98107+05   0.00000+00
       1.00000+06   0.00000+00
       2.51189+06   0.00000+00
       6.30957+06   0.00000+00
       1.58489+07   0.00000+00
       3.98107+07   0.00000+00
       1.00000+08   0.00000+00
```

Figure 6(a)
NET-2 Printed Response Output

NET-2 is very flexible in the models it supplies. Bipolar transistors, diodes, tunnel diodes, junction FET's, MOSFET's, transmission lines are some of the built-in models. The program does suffer from inadequate documentation.

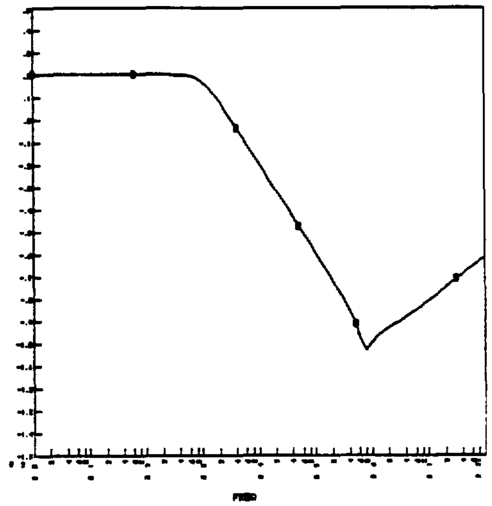


Figure 6(b)
SPICE-2 Plot Output

## Recent Directions and the Future for Circuit Analysis Programs

The second generation circuit analysis programs do a very good job of predicting the transient response of nonlinear electronic circuits. Most of the problems of the earlier programs have been solved with improved integration methods, sparse matrix techniques, new analysis formulations, and improved modeling.

Numerical techniques continue to be improved and incorporated into the latest versions of programs. Frequently, though, good software engineering practices are not followed. Examples are the total (almost) lack of comments in NET-2, the lack of good modularization and variable naming in SCEPTRE, improper use of variable typing in SPICE2, and the user input format unfriendliness of EMTP. Much more attention needs to be given to software engineering in future analysis program work.

Strong arguments can be made for the use of circuit analysis programs. Their use in simulating integrated circuits before physical implementation in silicon is firmly established. Safety analyses and design reviews are also frequent use of analysis programs. Electromagnetic radiation vulnerability analyses could become an increasingly important use of circuit analysis programs.

The size of circuits to be analyzed has always challenged circuit analysis programs. Early programs had problems in integration routines (slow) and storing circuit sizes (insufficient memory). Integration method improvements and sparse matrix techniques solved these problems. However the advent of LSI and VLSI circuits are again challenging capacities and speeds. New numerical techniques may come to the rescue but more probably the solution lies in hierarchical modeling and analysis.

## References

1. Kron, G., Tensor Analysis of Networks, New York, 1939.

2. Branin, Jr., F. H., "dc and Transient Analysis of Networks Using a Digital Computer," IRE International Convention Record, 10, Part 2, 1962, pp. 236-256.

3. Branin, Jr., F. H. "Computer Methods of Network Analysis," Proc. IEEE, 55, November 1967, pp. 1787-1801.

4. Johnson, E. D., C. T. Kleiner, L. R. McMurray, E. L. Steele, and F. A. Vassallo, Transient Radiation Analysis by Computer Program (TRAC), Autonetics Div., Rockwell International, Harry Diamond Laboratory Technical Report, June 1968.

5. "Automated Digital Computer Program for Determining Responses of Electronic Systems to Transient Nuclear Radiation (PREDICT)," IBM Space Guidance Center, Oswego, N.Y., IBM File 64-521-5, July 1964.

6. "1620 Electronic Circuit Analysis Program (ECAP)," IBM Corporation Data Processing Division, White Plains, N.Y., Applications Program 1620-EE-02X, 1965.

7. Malmberg, A. F., F. L. Cornwall, and F. N. Hofner, "NET-1 Network Analysis Program," Rept. LA-3119, Los Alamos, N.M., 1964.

8. Milliman, L. D., W. A. Massena, and R. H. Dickhaut, "CIRCUS--A Digital Computer Program for Transient Analysis of Electronic Circuits-User's Guide," Boeing Co., Harry Diamond Laboratory, July 1971.

9. Gear, C. W., "The Automatic Integration of Stiff Ordinary Differential Equations," in Information Processing 68, A. H. H. Morell, ed., North-Holland, Amsterdam, 1969, pp. 186-193.

10. Tinney, W. F., and J. W. Walter, "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization," Proc. IEEE, 55, Nov. 1967, pp. 1801-1809.

11. Branin, Jr., F. H., "ECAP-II--A New Electronic Circuit Analysis Program," IEEE J. Solid State Circuits, SC-6, August 1971, pp. 146-166.

12. "Automated Digital Computer Program for Determining the Response of Electronic Systems for Transient Nuclear Radiation (SCEPTRE)," Air Force Weapons Laboratory Technical Report AFWL-TR-69-77, Kirkland AFB, N.M., 1969.

13. Becker, David, et al., "SCEPTRE (Improved)," USAF Technical Report AFWL-TR-73-75, Air Force Weapons Laboratory, Kirkland AFB, N.M., 1973.

14. "Advanced Statistical Analysis Program (ASTAP)," IBM Corp-Data Processing Division, White Plains, N.Y., Installed User Program, SH-20-1118-X.

15. Malmberg, A. F., "User's Manual--NET-2 Network Analysis Program," Harry Diamond Laboratory Report AD-752-600, Braddock, Dunn, and McDonald, September 1972.

16. Dembart, B., and L. Milliman, "CIRCUS-2--A Digital Computer Program for Transient Analysis of Electronics Circuits--User's Guide," Boeing Co., Harry Diamond Laboratory, July 1971.

17. Nagel, L. W., and D. O. Pederson, "SPICE Electronics Research Laboratory Report," College of Engineering, University of California, Berkeley, CA, April 1973.

18. Chua, L. O., and P-M Lin, Computer-Aided Analysis of Electronic Circuits, Prentice-Hall, Englewood Cliffs, New York, 1975.

19. "Electromagnetic Transients Program (EMTP)," User's Manual, Bonneville Power Administration, Portland, Oregon, November 1977.

## DISCLAIMER

**This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government thereof, and shall not be used for advertising or product endorsement purposes.**